

Using Virtual Reality for Teaching Solid Geometry

A Case Study for a Cube Section

Alena Kovárová, Michal Sokolský

Institute of Applied Informatics
FIIT, Slovak University of Technology in Bratislava
Bratislava, Slovakia
kovarova@fiit.stuba.sk

Abstract— In recent years virtual reality has found application in e-learning. It provides interesting and attractive view of subject taught and students are more interested in learning. In this paper we propose the method of using virtual reality in desktop application that is intended to be used for solid geometry teaching at high schools. Except of the problem of displaying space (we used anaglyph technology) we paid attention to a problem of collision detection between two objects in 3D space with exact calculation of collision point. Our main goal was to create a good user interface design that is simple enough for quick understanding of the application and its intuitive control even for the first usage.

Keywords: *Virtual Reality, Solid Geometry*

I. INTRODUCTION

There exist a lot of well-designed solutions for 2D Geometry (The Geometer's Sketchpad [1]; Geogebra [2], Cinderella [3]; Geometry Master 3.0 [4]), but dealing with Solid Geometry we need to move from 2D plane to 3D space. When working with virtual space, situation gets a little bit complicated. The most obvious examples of such complications are

- a) *if the space cannot be turned (for example it is a picture in a book), it can be difficult to imagine the drawn objects correctly* [5]
- b) *an object, which looks to be in the front, but it is in the back*
- c) *two lines which look to have an intersection, but they do not have it*
- d) *the more infinite objects (lines, planes, surfaces) are used, the more confusing can the whole scene be*

This is real problem, because these can discourage users (and often did), but we want to achieve the opposite. Majority of these problems can be solved simply by using 3D visualization, e.g., using head mounted displays as in project Construct3D [6]. In our research we stayed with 2D displays - we looked for different solutions which can help users (students as well as teachers) to work with abstract 3D object within 3D space displayed in 2D. This is mostly a matter of a good user interface. We propose a new application control for users working with 3D geometry on a 2D screen, mainly to reduce the problem with unknown depth (the third dimension which is not visible), and so make it easier for users.

II. 3D WORLDS

First of all, we studied different systems dealing with 3D space to analyze all types of control – examine their user interfaces and find out how they dealt with above mentioned typical problems. We determined 3 basic areas of applications dealing with 3D: 3D games, 3D graphical editors and educational software dealing with geometry. These systems can provide different functionalities of various 3D virtual worlds, we focused on their common features as well as their differences which helped us to realize the type of interactions, with which are users already familiar.

A. 3D games

This is very important area, because the majority of users (students) has skills mainly with this type of applications and therefore is familiar with their type of control. There are different types of games. Historical development divides games into those that simulate a 3D world, but a player can move only on the surface of this world (famous pioneers are Wolfenstein from 1992 [7] and Doom from 1993 [8]) and those where the player is able to move in all 6 degrees of freedom, e.g., Descent from 1995 [9]. In present 3D games dominate those that offer only 2D movements - on the surface. The demands for better control of the games are not probably as big as for better graphics, because even if we can now find more advanced control input devices such as a wii remote or a gamepad, there are still many games and players staying with the traditional mouse and keyboard control and the players are used to a very similar game controls as it was 15 years ago: Mouse movements can change the view point, a movement direction or simply moves the mouse focus. This can be often performed in combination with the keyboard – keys WASD and/or arrow keys. The left mouse button is mapped to the most often used action – to run an active object (to fire a weapon, to select an object, etc.), if there is already a selected object, it can designate an application object, which is similar to drag-and-drop style. The right mouse button is often mapped to the second mode of control, which is not as often used as the one on the left button or it simply provides a context menu. The wheel is often used for scrolling in selected menu or for zooming in and out. The other keys on the keyboard can be mapped to other game functions; which function is mapped to which key is dependent on how often the function is used and how far the key is from a basic hand position - usual basic hand

position is on WASD keys and therefore Q, E, R, F, Shift, Ctrl, Alt and Space keys are often included among the control keys.

B. 3D graphical editors

There are many different types of 3D editors. Some are designed for technical drawing (for engineers: CAD/CAM, rapid prototyping, etc.), 3D animation, others for art and design. Some, because of their broad functionality, require many hours of training, while other editors can be handled by a beginner. For simple comparison we have selected the 5 common editors, covering different categories: Blender [10], Autodesk® 3ds Max® [11], Autodesk® Maya® [12], SketchUp [13] and Pixologic ZBrush [14]. Blender is an open source editor similar to technical editors, but many of its functionalities are intuitive and thus easily accessible for beginners. 3ds Max and Maya are complex editors with a wide functionality, both for 3D animation (almost on the level of technical type), both proprietary. They differ in the specific features and even in some basic controls, since each of the editors was originally developed by a different company. SketchUp in its basic form is intended primarily for beginners, it is free and it is also one of the technical editors. ZBrush is, unlike the previous, an editor for artists, because the principle of objects is fundamentally different - it uses the metaphor of shaping clay, and is proprietary. The comparison shows that more technical editors provide 4 views, while one view is more understandable for beginners. Basic functions such as a function to control the view and the scene differ between editors, too. Of special interest is the fact, that every editor has three modes (translation, rotation, scaling) between which it is necessary to switch. Only Blender has implemented recognition of gestures and the system itself can automatically detect the mode (it is obviously slower than a keyboard shortcut, but very useful for beginners). These three modes are usually visually clearly distinguishable (see example in Figure 1). In all editors the operation of inserting a new object requires more interaction steps; the shortest way is to click on the toolbar.

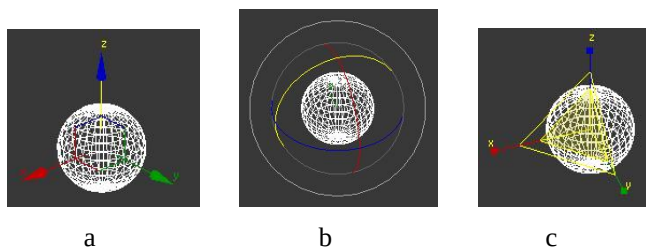


Figure 1. Different object illustration in different editing modes in 3ds Max editor – dragging (a), rotating (b) and scaling (c)

C. Educational geometry editors

There are different educational pieces of software working with 2D or 3D covering various mathematical areas [15]. Because our interest is mainly in editors that simulate three-dimensional drawing assignment, we did not pay attention to the editors that have as an input a function or other text input (such programmable software as Maple, Mathematica, MatLab or educational software using Logo and/or turtle graphics). We focused primarily on those that resemble drawing on a paper with a ruler and a pencil. This condition excludes also

a) *Geometria* [16] where the user is drawing by setting input from keyboard – e.g., names of points - and not by mouse (but it supports setting problems and then searching for solution or creating the standard one)

b) *Géoplan-Géospace* [17] whereas it requires a text input, often mathematical functions

c) *Geometer's Sketchpad* [1], *GeoGebra* [2], *Cinderella* [3] and *Geometry Master* [4] since they work only with 2D geometry (but they are well-designed)

d) *GeomSpace* [18] and *Mathsay* [19] where the user needs to work with digits – a lot of input fields

e) *Yenka* [20] contains a lot of interactive animations working with 3D geometry, but none working with drawing.

Finally the last two applications that meet our requirements are *Archimedes Geo3D* [21] and *Cabri 3D* [22], both proprietary. Both applications have a wide range of drawing - contain all the essential principle possibilities of working with elementary objects (tools), which are well-arranged and grouped in a toolbar (see Figure 2 and Figure 3).

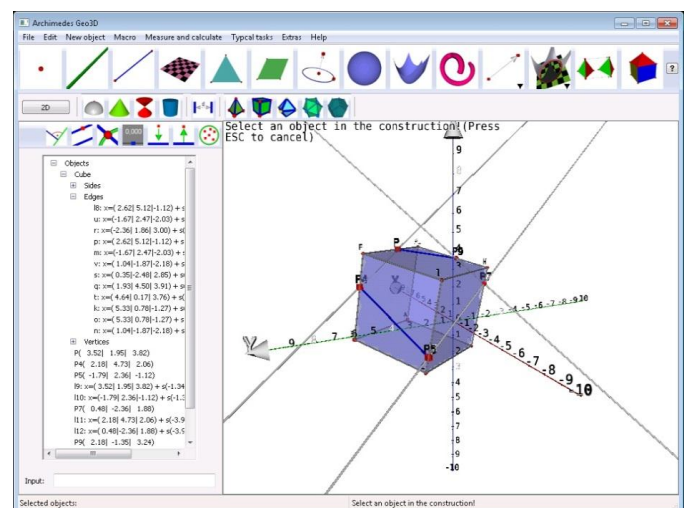


Figure 2. Screenshot of Archimedes Geo3D application

None of these two applications provides shortcuts to work with tools, thus the drawing consists of relentless clicking in the toolbar and back into the scene. Equally arduous is a selection of one of the grouped tools (for example group for adding a line: add a segment, prolonging an existing segment, add a line defined by two points or defined by a point and parallel line, etc.). Some of these can be automatically detected by the system itself, but the majority has to be chosen directly by the user. The next feature is a basic axial cross with numeric axes Archimedes Geo3D that on one side can assist in the orientation in space, on the other side may be constrained. The base (horizontal) plane in Cabri 3D can cause the same problem. Both applications have a problem with infinite objects (line, plane), because their size / length often make the whole scene unclear.

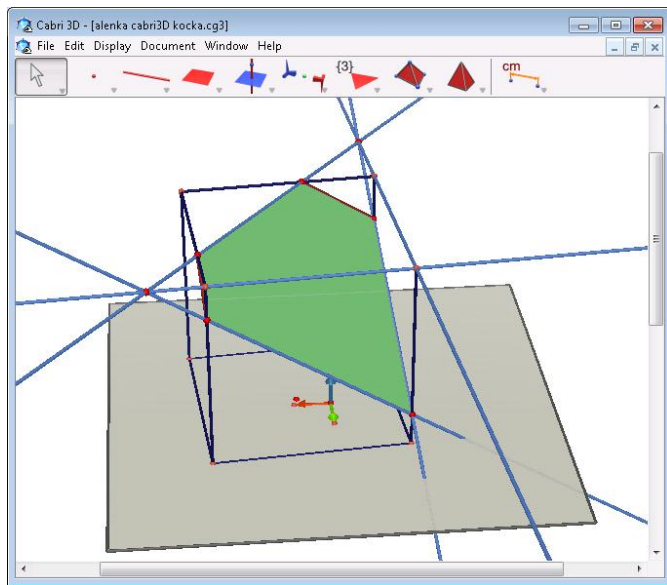


Figure 3. Screenshot of Cabri 3D application

Both applications provide hints during inserting a new object. In Cabri 3D it is implemented in form of a textbox on the mouse position (see Figure 4), the hovered object changes its color and starts to vibrate. According to the mouse position (which object it hovers above/what is possible to do) the content of hint changes – it informs the user what can be done at the current mouse position.

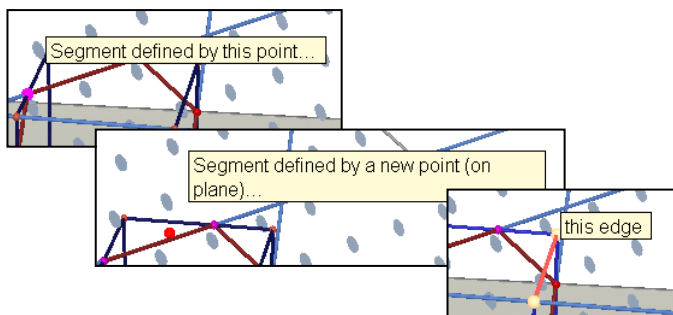


Figure 4. Examples of hints for adding a new segment in Cabri 3D

In Archimedes Geo3D the simple text hint is at upper left corner of canvas, the same hint is at status bar. Next to the mouse position the label of the hovered object is displayed. If the addition of a new object requires more than one click (e.g., a new segment), the system does not show a preview of user's action (e.g., the segment which ends at the mouse position until the second click). Moreover, the selected tool-button is not visually emphasized (thus the user cannot be sure if it is really pressed = missing feedback). Also neither hovered nor selected object in the scene is visually emphasized. These features are significant usability deficiencies of Archimedes Geo3D user interface.

Although Cabri 3D has better user interface, it does not work very well with drag-and-drop interaction style (neither does Archimedes Geo3D). By drag-and-drop the user can

zoom or rotate the object, or rotate the scene, or make the dynamic change, e.g., move a point. But it does not work when the user wants to extend or shorten an object. Another negative feature is impossibility to simply interact with overlapped object. The user has to rotate the scene, sometimes even zoom it in (or use another complicated bypass) to get rid of this overlap.

III. INTUITIVE INTERFACE AND CONTROL OF VIRTUAL 3D SPACE

For four above mentioned problems we propose following solutions:

- The scene can be rotated (that works in each 3D application but not in each educational material even if digital)*
- The further the object is, the more darker/brighter it is (some applications use perspective but in this case the user can lose the sense for collinearity, which can be in some case crucial), sometimes works instead of color change the thickness/size or dash/fill type.*
- When the user tries to interact with seeming intersection, the scene turns itself a little bit, this can help the user to see, that the lines are nonintersecting.*
- Infinite objects are not rendered in their real size to the edge of the screen/window, but only the small, important part of them. The user has later on a possibility to prolong them in any direction, if needed.*

And for a better educational impact, it can contain a function, which can check the correctness of the student's solutions (whether the assignment was created by a teacher, by a student or is a default one from the system) or even can be a part of simple learning management system.

To verify our ideas (especially the third and the forth) it has to be created an example, which can be tested. From our point of view, it is not important which case is chosen, because the same principles will work in any other. We chose a cube cross section, because it can help students with lower spatial ability [5] to go through it.

For this purpose we created an application named Stereo3D (see Figure 5). Here we used our previous experience of project StereoVR Portal [23]

Stereo3D basic description is:

The basic scene contains a cube with three points on it, which defines the crossing plane. The user can through menu or context menu do these actions:

- add a new point (only if it is an intersection) or delete it,
- add a line (connecting two points or a parallel one anchored to a point),
- delete, prolong or shorten the line (see Figure 6)
- define a cross section
- let the system check if the cross section is correct

- force the system show the cross section (as a hint)
- turn on/off anaglyph projection

For a mouse is mapped interaction as follows:

- Left mouse button: press button in menu (toolbar or context menu), select an object, and translate the scene
- Right mouse button: rotating the scene
- Mouse wheel: zooming in/out the scene,
- No button pressed on mouse: translating a selected object (if there is any), highlight a hovered object

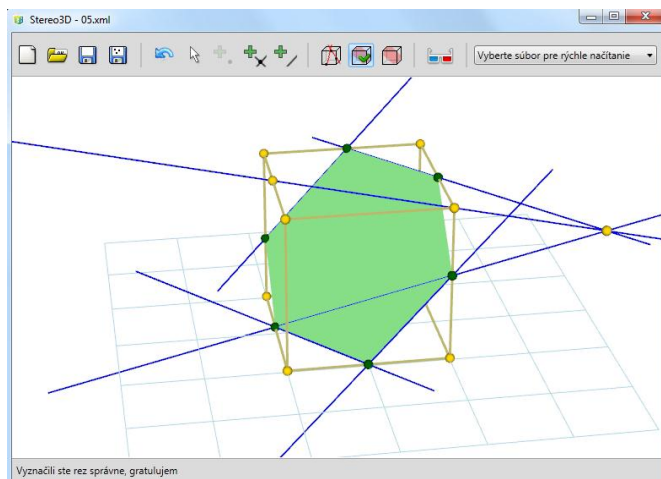


Figure 5. Screenshot of our application Stereo3D

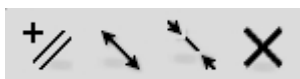


Figure 6. Context menu for a line

IV. 4 EVALUATION

During the continuous development of our application, we tested different parts of our interaction design between our friends. We also consulted it with a professional teacher - Mgr. Iveta Kohanová, PhD. who is Head of Mathematics at the Department of Algebra, Geometry and Mathematics Education, Faculty of Mathematics, Physics and Informatics Comenius University in Bratislava, Slovakia and few other teachers from high school. Finally we prepared a test for evaluation. Here we had two different groups of testers. The first one consists of high school students in their first year at high school Grösslingová 18 in Bratislava (28 students) and the second group were future teachers, students, who studied on a bachelor degree at the Faculty of Mathematics, Physics and Informatics Comenius University in Bratislava (11 students). There has to be said, that high school students did not pass yet the curriculum containing cube cross sections, which was also seen in some of the feedback questionnaire responses, but despite this their work was very good with the application.

Before starting the test on high school we explained to the students the purpose and objectives of the application testing. Since they had not studied that curriculum yet, we explained it to them in short, so they at least have the basic idea and can better understand what was expected in our application. We prepared for them a few simple assignments on cube section (e.g., the section plane was defined by adjacent cube corners). Students were told how the entries in the application loaded and how they can check the accuracy of solutions and show the correct solution if they have no idea how it should look like. They were not told anything else. They got a few minutes to try to solve the examples and then each student got the feedback questionnaire (see next subsection)

Testing future teachers was similar. However, since they have sufficient knowledge of sections of cubes, they just received information about how to load an assignment in our application. The prepared assignments were different from previous group, their difficulty was incrementally increasing (up to the most difficult one: points on three non-neighboring edges of the cube). The aim of students was to work with application from the teacher's perspective, thus they weren't provided with further information and they had to find out themselves how to control the application. They got a few minutes to test applications and examples of solution and subsequently received the same questionnaires as high school students.

A. Questionnaires

The feedback questionnaire was divided into four areas:

- Look & Feel* – how do they like it
- Arrangement* – a good arrangement of elements is important so the user recognizes the features of the program without the assistance and it has to be easy to navigate.
- Controls* – they are the most important part from the user's point of view. If application has problematic controls or the user does not know or understand how to control it, it has lost every chance for further using.
- Status Bar* – its aim was to provide on-going information and application feedback to the user. To make it more noticeable, the status bar flashed every time its information was changed. For this reason, one of the questions was if the users noticed this flashing (noticed new information) and whether this flashing does not disturb them in their work.

The majority of questions offer answers scaled to a scale from 1 to 5 (grade), where 1 usually means the fulfilled expectation. Some of the questions offer only the yes/no answer, which we mapped on 1 and 5. If there was a different answer pair, we mapped it on 2 and 4. For the rare question of type: "What would you change?" there was an empty field (no numerical expression for this type).

Since our graph is not easy to read, we decided to express the test results in quickly understandable form – we represent each answer v_j by value calculated as weighted average of all responses in test group:

$$v_j = \frac{\sum_{i=1}^5 i \cdot a_{i,j}}{\sum_{i=1}^5 a_{i,j}}$$

where v_j is value of j^{th} answer for the whole test group, i goes through all possible grades that can have a single answer from a single tester and $a_{i,j}$ is the number of testers, which graded their j^{th} answer by the grade i . Since i goes from 1 to 5, the closer is the value v_j to 1, the result is better, closer to 5, the worse.

For easier handling, we have created for our three groups following abbreviations: High school students – HS, future teachers – FT.

B. Test results

Our prototypes were accepted positively. The users considered it simple, intuitive and user friendly (HS: 2.3, FT: 1.6). The future teachers showed a big interest in it. The ability to check the correctness of the solution and the ability to demonstrate the correct solution was evaluated very positively.

The total number of high school students was 28. For most of them our application was comfortable (HS: 2.0), clear, with appropriately spaced buttons (HS: 1.7). Their meaning constituted a problem for a few students, but it can be attributed to the fact that they have heard about the cross section for the first time just before starting the application, so they were not familiar with the construction method. Students considered program controls very nice (HS: 2.0) and intuitive (HS: 2.2), and with a few exceptions no one had any problem with adding objects to the scene. Rotating and moving scenes were evaluated very positively (HS: 1.3), and except for two students, the assignment of these functions to the right and left mouse button suited them. Quite a lot of students had problems to carry out the intended operation in the application (HS: 2.7), which may in part be attributed to the fact that they were not able to learn the cube section in the short time they had available for testing. We believe that the longer test application would reduce the percentage. More than half of the students noticed the status bar during the work (HS: 2.5) and over a third noticed the help information (HS: 2.5). The vast majority was not distracted by it during their work (HS: 4.6).

We had 11 university students – future teachers. They evaluated the application very positively (FT: 1.6). Almost all elements of the application worked nicely and clearly (FT: 1.5), the buttons were conveniently arranged (FT: 1.5) and there was no problem to understand it (FT: 1.1). Controls were intuitive and comfortable (FT: 1.3), the only exception was adding objects to the scene. Shifting and rotation of the scene was natural and fully complied with the use of the left and right buttons (FT: 1.0-1.5). We can negatively evaluate the fact that almost a third of respondents reported that they sometimes felt that they could not perform the intended operation (FT: 3.8). The origin of this problem may be in an additional pop-up menu that appears when the user point at an object in the scene after a short interval. One solution might be the reduction of this interval, which, however, could cause too frequent menu popping up when moving the mouse in the scene. The status bar was noticed by three quarters of respondents (FT: 1.8) and

the vast majority indicated that it was not distracting at all (FT: 1.6). Information was seen helpful only by about a third of respondents (FT: 3.4), which is possibly caused by their education and background.

C. Proposals for improvement and change

Students had in questionnaire the opportunity to freely express their views on the application for what would it have changed, or how it would improve. The most interesting remarks were:

a) *Additional pop-up menu as part of the main menu* - Location of all available keys to the main menu would increase transparency and ease of application and in addition would make the operation of objects, it would be necessary to additionally indicate the object with which the user intends to interact.

b) *Additional popup menu on mouse click* - From previous tests we know that if there would be context menu which would appear only after a mouse click on an object, a large proportion of users would not even found it. Now it appears alone, it comes out when the mouse cursor hover an object that provides this menu.

c) *Longer lines* - A relatively large number of prospective teachers in the questionnaire said they would welcome longer lines, because now they must often be extended.

d) *A button for centering a view* - This is a very good idea that can help users set the camera to the initial state, which looks exactly the cube.

V. CONCLUSION

As the result shows, we created application, which has intuitive interface and therefore was easy to understand and easy to use. Unlike other similar applications, in our application are also shortcuts, so the user does not need to move a mouse to menu and back to scene every time when he (she) wants to replace the tool. Our application eliminates the problem with false intersections. We shorten the infinite objects (lines), which were considered as too short, but it makes the scene more organized; if needed, any line can be prolonged. For real 3D sense we implement also anaglyph visualization but in such simple scene it was not necessary to turn it on. According to math teachers, the most valuable feature is the ability of automatic evaluation of student's solution correctness, which they never met in previously known applications.

ACKNOWLEDGMENT

This work was partially supported by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. KG 244-022STU-4/2010

REFERENCES

- [1] Jackiw, N. (2009). The Geometer's Sketchpad v5.0. Emeryville: Key Curriculum Press.

- [2] International GeoGebra Institute. (2011). Geogebra. Retrieved 2011, from <http://www.geogebra.org/>
- [3] Kortenkamp, U. H., & Richter-Gebert, J. (2000). User Manual for the Interactive Geometry Software Cinderella. Heidelberg: Springer.
- [4] Caltrox Educational Software. (2011). Geometry Master 3.0. Retrieved 2011, from Caltrox Educational Software: <http://www.caltrox.com/products/geometry.htm>
- [5] Cohen, C. A., & Hegarty, M. (2007). Sources of Difficulty in Imagining Cross Sections of 3D Objects. The 29th Annual Conference of the Cognitive Science Society. Nashville, Tennessee, USA.
- [6] Kaufmann, H., & Schmalstieg, D. (2002). Mathematics and geometry education with collaborative augmented reality. ACM SIGGRAPH 2002 conference abstracts and applications (pp. 37-41). San Antonio, Texas: ACM.
- [7] Id Software LLC. (2002). Wolfenstein 3D. Retrieved 2011, from Id Software: <http://idsoftware.com/games/wolfenstein/wolf3d/index.html>
- [8] Id Software. (1993). Doom 3. Retrieved 2011, from Id Software: <http://idsoftware.com/games/doom/doom3/index.html>
- [9] Interplay. (1995). Descent. Retrieved 2011, from Interplay: <http://interplay.com/games/descent.php>
- [10] Diz, O. V. (2010). Blender 2.5 - Basic Interface. Retrieved 2011, from www.blendnuts.com: <http://www.blendnuts.com/2010/06/blender-25-interface.html>
- [11] Autodesk, Inc. (2010). Autodesk 3ds Max 2011. Retrieved from http://images.autodesk.com/adsk/files/3ds_max_2011_tutorials.pdf
- [12] Autodesk, Inc. (2010). Autodesk Maya 2011 - Getting Started. Retrieved from http://images.autodesk.com/adsk/files/3ds_max_2011_tutorials.pdf
- [13] Google. (2011). New to Google SketchUp - Video Tutorials. Retrieved 2011, from http://sketchup.google.com/intl/en/training/videos/new_to_gsu.html
- [14] Pixologic, Inc. (2011). Getting Started with ZBrush guide. Retrieved 2011, from <http://download.pixologic01.com/download.php?f=Plugins/zbrush4->
- [15] CIGS. (1994-2011). Interactive Geometry. Retrieved 2011, from The Math Forum: <http://mathforum.org/dynamic/classroom.html>
- [16] Dumitrascu, S. (2000-2011). Geometria - Interactive Geometry Software . Retrieved 2011, from <http://www.geocentral.net/geometria/en/>
- [17] Aid-creem. (1994-2000). Geoplan-Geospace. Retrieved 2011, from Association pour l'Innovation Didactique: <http://www.aid-creem.org/geoplace.html>
- [18] GeomSpace. (1999-2011). Retrieved 2011, from <http://sourceforge.net/projects/geospace/>
- [19] Golden Oct. Computer Network Service Co., Ltd. (?-2008). MathSay. Retrieved 2011, from <http://www.mathsay.com/>
- [20] Crocodile Clips. (2000-2011). Retrieved 2011, from Yenka: <http://yenka.com/>
- [21] Goebel, A. (2008). Documentation for Archimedes Geo3D. Retrieved July 14, 2011, from raumgeometrie.de: <http://raumgeometrie.de/>
- [22] de Cotret, S., & de Cotret, P. R. (2005). Cabri 3D User Manual. Cabrillog.
- [23] Kovárová, A., & Gregor, J. (2009). Solid Geometry and Virtual Reality. SCO 2009, Proceedings of 6th conference on electronic support of education, (pp. 93-98). Brno.